



EDITORIAL

Guidelines for writing software building reports

Pautas para la redacción de informes de creación de softwares

Annier Jesús Fajardo Quesada¹  , René Herrero Pacheco¹  

¹Universidad de Ciencias Médicas de Granma. Facultad de Ciencias Médicas de Bayamo. Granma, Cuba.

Cite as: Fajardo Quesada AJ, Herrero Pacheco R. Guidelines for writing software building reports. Gamification and Augmented Reality. 2024; 2:39. <https://doi.org/10.56294/gr202439>

Submitted: 22-10-2023

Revised: 04-02-2024

Accepted: 23-04-2024

Published: 24-04-2024

Editor: Adrián Alejandro Vitón Castillo 

ABSTRACT

The linkage of software with various areas of knowledge has increased significantly, highlighting the importance of process automation thanks to the expansion of imaginative capabilities in this area. The concept of software now encompasses applications on different operating systems, including PWAs, websites, command line applications and APIs. This landscape demands standardized methods for the creation and dissemination of applications, although there is currently no detailed guidance for writing reports on software creation. It is crucial to adopt methodological guidelines that allow the reproducibility of software research, clearly differentiating between the creation and the evaluation-validation of these tools. Such differentiation helps to avoid confusion and ensures consistency in reporting, especially when the objective is to develop new tools. Before starting software development, it is essential to consider the objective, scope and existence of similar technologies, assessing the real need to create a new solution. The article proposes a detailed outline for writing software development reports, covering from the introduction, method, results, to the discussion and conclusions. This outline covers aspects such as the theoretical and historical framework, the importance and necessity of the product, classification of the study, details of the creation process, technologies used, and software evaluation. It highlights the importance of performing quality tests throughout the development, ensuring that the final results reflect the most recent and conclusive state of the software. The creation of software is presented as a non-linear process that requires iterations and adaptations based on functionality and quality tests, whose results directly influence the progress of the project. The suggested model is flexible, allowing adjustments according to the specific characteristics of the software under development.

Keywords: Process Automation; Software Development; Research Methodologies; Development Reports.

RESUMEN

La vinculación de software con diversas áreas del conocimiento ha aumentado significativamente, destacando la importancia de la automatización de procesos gracias a la expansión de las capacidades imaginativas en este ámbito. El concepto de software ahora engloba aplicaciones en distintos sistemas operativos, incluyendo PWA, sitios web, aplicaciones de línea de comando y APIs. Este panorama demanda métodos estandarizados para la creación y divulgación de aplicaciones, aunque actualmente no existe una guía detallada para la redacción de informes sobre la creación de software. Es crucial adoptar pautas metodológicas que permitan la reproducibilidad de la investigación en software, diferenciando claramente entre la creación y la evaluación-validación de estas herramientas. Dicha diferenciación ayuda a evitar confusiones y garantiza la coherencia en los informes, especialmente cuando el objetivo es desarrollar nuevos instrumentos. Antes de comenzar con el desarrollo de software, es esencial considerar el objetivo, alcance y la existencia de tecnologías similares, evaluando la necesidad real de crear una nueva solución. El artículo propone un esquema detallado para la redacción de informes de creación de software, cubriendo desde la introducción, método, resultados, hasta la discusión y conclusiones. Este esquema abarca aspectos como el marco teórico

e histórico, la importancia y necesidad del producto, clasificación del estudio, detalles del proceso de creación, tecnologías utilizadas, y evaluación del software. Resalta la importancia de realizar pruebas de calidad a lo largo del desarrollo, asegurando que los resultados finales reflejen el estado más reciente y concluyente del software. La creación de software se presenta como un proceso no lineal que requiere iteraciones y adaptaciones basadas en pruebas de funcionalidad y calidad, cuyos resultados influyen directamente en el avance del proyecto. El modelo sugerido es flexible, permitiendo ajustes según las características específicas del software en desarrollo.

Palabras clave: Automatización de Procesos; Creación de Software; Metodologías de Investigación; Informes de Desarrollo.

In the last few years, the link between software and all areas of knowledge has grown vertiginously. The fact that imagination is the limit of the processes that can be automated has increased the number of software researchers, especially researchers dedicated to creating them. Software today encompasses any application regardless of the target operating system, whether installable or portable and includes PWAs (progressive web applications) and websites, as well as command line applications and APIs (Application Programming Interfaces).

Several methods standardize an optimal way to create applications,⁽¹⁾ both in general and specific to the field or type of application, but science does not stop at creation; the next step is dissemination, and for this, written reports are used, and one of the most popular types is the scientific article format. A detailed guide for writing software creation reports needs to be provided.

Research without method is a chaotic science which cannot be reproduced, so it is necessary to establish methodological guidelines for such research and adapt it to the publishable format for further dissemination.

It is always advisable to separate the creation and evaluation-validation reports for software research reports since each is an independent research environment linked by the same product but with different techniques. This does not mean that the report of the creation process does not refer to the results of the tests carried out to determine the quality of some functionalities.

A report that aims to create a new tool needs new specificities because its method coincides with the results, which can lead to clarity and coherent or redundant wording.

Before starting the creation of any software and its subsequent report, the objective and scope of the same should be taken into account after a previous investigation of whether similar technologies exist so as not to create duplicates, as well as whether it is really necessary to create a software and not a format readable by another technology, for example, a website or web application on a particular topic for students of a university when the same information can be given in an interactive PowerPoint and distributed through social networks.

An outline that can be followed for writing software creation reports is as follows:

Introduction

- Theoretical framework: Main concepts
- Historical framework: What was it like before the app performed the tasks it automates or executes?
- Importance of the product: What problem does it solve?
- Product need: Is the product unique in the 'world', and if not, how does it outperform its peers (feasibility, cost, efficiency, design, etc.)?
- Objective

Method

- Classification of the study
 - Usually, technological innovation studies
- General data of the creation process
 - A place where the software was created
 - Size of the work team and experience of its members
 - The time it took to create the software
 - Other data of interest
- Software development method, if any, used
- Programming languages
- Modules or packages of the languages and the purpose of each one
- Tools and technologies
 - FrontEnd or UI (User Interface) design.
 - Databases

- Storage
 - Binding with APIs
 - Other
- System on which it was mounted
 - Operating system and kernel and its versions (if applicable)
 - Memory
 - Processor
- Method for collecting the information (in the case of products that need to display information or databases, among others) Proceed as for review articles
 - Databases and search engines
 - Search terms
 - Filters
 - Methods of information analysis
- Software evaluation method
- Statement of ethical principles, if necessary

Results

- Describe UI
 - Show images of the functional app, representative Windows
- Describe UX (User Experience)
- What utilities were added to improve the user experience, e.g. Tooltips, Shortkeys, etc?
- Describe final requirements, necessary dependencies
 - Hardware
 - Operating System
 - Applications
- Results in the last test evaluation
 - Speed
 - Graphic Interface
 - Efficiency
 - Accessibility

Discussion

- The results contrast with other studies of similar applications or with the official documentation of such applications, highlighting where it exceeds and where it is surpassed.

Conclusions

- Strengths and limitations
- Projections and expected scope

Remember that creating software or applications is not a linear process, and quality or functionality tests must be performed to determine the progress or regression of the process. The report must include the last results because they are the conclusive ones in the creation phase. These tests are not definitive because validation studies demonstrate the app's scope in other environments.

Depending on the type of software, some of the points presented in this model can be extended or simplified, as some are only of visual content, others with purely analytical elements or even a combination of both. Adding other data as appropriate is also valid since it is a flexible model.

REFERENCES

1. Carrizo D, Alfaro A, Carrizo D, Alfaro A. Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Ingeniare Rev Chil Ing* [Internet]. 2018 [citado 6 de febrero de 2024];26(1):114-29. Disponible en: <http://dx.doi.org/10.4067/S0718-33052018000100114>

FINANCING

None.

CONFLICT OF INTEREST

None.

AUTHORSHIP CONTRIBUTION

Conceptualization: Annier Jesús Fajardo Quesada, René Herrero Pacheco.

Research: Annier Jesús Fajardo Quesada, René Herrero Pacheco.

Visualization: Annier Jesús Fajardo Quesada, René Herrero Pacheco.

Writing - original draft: Annier Jesús Fajardo Quesada, René Herrero Pacheco.

Writing - revision and editing: Annier Jesús Fajardo Quesada, René Herrero Pacheco.